Comput. Math. Applic. Vol. 15, No. 5, pp. 367–373, 1988
Printed in Great Britain.

0097-4943/88 \$3.00 + 0.00

AN ACCELERATED SUCCESSIVE ORTHOGONAL PROJECTIONS METHOD FOR SOLVING LARGE-SCALE LINEAR FEASIBILITY PROBLEMS

J. M. Martínez
Departamento de Matemática Aplicada, IMECC–UNICAMP, Caixa Postal 6065, 13081, Campinas SP, Brasil

(*Received* 16 *September* 1987)

Communicated by E. Y. Rodin

Abstract—We consider linear feasibility problems in the "standard" form $\mathbf{Ax} = \mathbf{b}, \mathbf{l} \leqslant \mathbf{x} \leqslant \mathbf{u}$. The successive orthogonal projections method may be used for solving this problem using sparse orthogonal factorizations techniques for computing the projections on $\mathbf{Ax} = \mathbf{b}$. We introduce an acceleration technique in order to speed up the (generally slow) convergence of the method. We present some numerical experiments.

1. INTRODUCTION

Many practical situations require the solution of large-scale linear feasibility problems:

Find $\mathbf{x} \in \mathbb{R}^q$, such that

$$\mathbf{A}_1\mathbf{x} \leqslant \mathbf{b}_1,$$
$$\mathbf{A}_2\mathbf{x} = \mathbf{b}_2,$$
$$\mathbf{l} \leqslant \mathbf{x} \leqslant \mathbf{u}. \tag{1}$$

In fact, the feasibility phase of primal algorithms for solving linearly constrained optimization problems [1, 2] is a problem of type (1). Generally, simplex-like techniques [1–4] are used to find the solution of (1). However, the existence of problems in which the matrix of constraints is sparse but with no special structure pattern, encouraged the development of row-action (projection) methods for handling this situation [5–13].

The main feature of row-action methods is that they are iterative procedures which, without making any changes in the original matrix, use their rows, one row at a time. Generalizations and improvements of row-action methods received substantial attention in the last years [9, 13–21].

The speed of convergence of most row-action methods is very low, and therefore, they are not considered to be competitive with simplex-like techniques, in the situation in which these techniques are applicable.

There exist several ways to improve the efficiency of projection methods without destroying their main characteristics. One of them is to project onto blocks of constraints instead of individual constraints [15, 22]. However, a close formula for projecting onto a set defined by several inequality constraints does not exist. This difficulty leads to a formulation "with slack variables" of the problem:

Find $\mathbf{x} \in \mathbb{R}^n$, such that

$$\mathbf{Ax} = \mathbf{b},$$
$$\mathbf{l} \leqslant \mathbf{x} \leqslant \mathbf{u}, \tag{2}$$

which, of course, is equivalent to (1), if $\mathbf{A}, \mathbf{x}, \mathbf{l}, \mathbf{u}$ are defined in a proper way.

Moreover, recent research on least-squares methods and other algorithms which need the calculation of projections [23–28] led to a notable improvement of techniques for computing projections on linear manifolds. In this line, symbolic factorization schemes permit to obtain the structure of the Choleski factorization of $\mathbf{AA}^\mathrm{T}$ previously to computing it [24–26]. A row-action-type method based on block-projections on the two sets defined in (2) may have some advantages over simplex-like techniques for large-scale linear feasibility problems. First, the code which implements a method of that class is rather simple, in comparison with advanced simplex algorithms. Second, the storage used by the whole calculation is known in advance, that is, fill-in doesn't depend on numerical computations. Because of these reasons, such a method should be very suitable for implementation in a microcomputer environment.

However, the speed of convergence of block-projection method is still expected to be very low. As a consequence, some authors proposed acceleration procedures (for linear and nonlinear systems) [18, 21]. In this paper we propose a highly opportunistic acceleration device for the application of a block-projection method to (2). We present a limited but very encouraging set of numerical experiments. Finally, we suggest a related scheme to handle general linear programming problems.

2. THE PROPOSED METHOD

Let $\mathbf{A}$ be a real, full-rank, $m \times n$ matrix, $m \leqslant n$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$. We consider the linear feasibility problem in the form (2).

Suppose $\mathbf{A}$ is stored columnwise. We obtain the columnwise structure of the lower-triangular matrix $\mathbf{L}$ such that $\mathbf{AA}^\mathrm{T} = \mathbf{LL}^\mathrm{T}$ using a symbolic manipulation scheme [24]. (In some situations, a columnwise structure which contains the structure of $\mathbf{L}$ may be obtained by direct inspection of the structure of $\mathbf{A}$.) Then, matrix $\mathbf{L}$ is obtained using the scheme of [24].

Let $\mathbf{x}^0$ be an arbitrary initial point $\mathbf{l} \leqslant \mathbf{x}^0 \leqslant \mathbf{u}, \epsilon_1, \epsilon_2, \epsilon_3$ small positive numbers, $0 < t < 1$ ($t \approx 0.1$), $0 < r < 2$. Given $\mathbf{x}^k$, the kth approximation to a solution of (2), we obtain $\mathbf{x}^{k+1}$ as follows:

Algorithm 2.1

Step 1—Compute $\mathbf{F}_k = \mathbf{Ax}^k - \mathbf{b}$. If $\|\mathbf{F}_k\|_\infty < \epsilon_1$, declare "Convergence of Type 0" and stop. Otherwise let s_k be such that the absolute value of f_{s_k} (the sth component of $\mathbf{F}_k$) equals $\|\mathbf{F}_k\|_\infty$.

Step 2—If $k = 0$, or $s_k \neq s_{k-1}$, or $|\alpha_k| \geqslant 1$, or $|\alpha_k - \alpha_{k-1}| > t$, with $\alpha_k = f_{s_k}/f_{s_{k-1}}$) go to Step 5.

Step 3—(Acceleration Step). Set

$$\mathbf{x}^k \leftarrow (\mathbf{x}^k - \alpha_k \mathbf{x}^{k-1})/(1 - \alpha_k). \tag{3}$$

Substitute $\mathbf{x}^k$ for its projection on the prisma $\mathbf{l} \leqslant \mathbf{x} \leqslant \mathbf{u}$

(for $i = 1$ to n,

if $x_i^k > u_i, x_i^k \leftarrow u_i$

if $x_i^k < l_i, x_i^k \leftarrow l_i$).

Step 4—Recalculate $\mathbf{F}_k, s_k$.

Step 5—(Projection on the linear manifold)

$$\mathbf{y} \leftarrow \mathbf{x}^k - \mathbf{A}^\mathrm{T}(\mathbf{LL}^\mathrm{T})^{-1}\mathbf{F}_k.$$

Step 6—Set $\mathbf{z}$ = the projection of $\mathbf{y}$ on the prisma $\mathbf{l} \leqslant \mathbf{x} \leqslant \mathbf{u}$.

Set $\omega = \mathbf{y} + r(\mathbf{z} - \mathbf{y})$, and

$\mathbf{x}^{k+1} \leftarrow$ the projection of ω on the prisma.

If, for all $i = 1, \ldots, n$,

$|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k| \leqslant \epsilon_2 |\mathbf{x}_i^k| + \epsilon_3$,

then declare "Convergence of Type 1" and stop. Otherwise, $k \leftarrow k + 1$. Go to Step 1.

Let us explain the motivation of the Acceleration Step 3. We know, from [29], that a geometric rate of convergence is expected for the (nonaccelerated) successive orthogonal projections method.

Now, at the final stages of a convergent SOP sequence, we expected that projections on inequality constraints involve only those constraints which are active at the limit point $\mathbf{x}^*$. Therefore, at those stages, a SOP method tends to take the form $\mathbf{x}^k - \mathbf{x}^* = G(\mathbf{x}^{k-1} - \mathbf{x}^*)$, for a suitably defined matrix. Thus, following [18, 30], we expect that

$$\mathbf{x}^k - \mathbf{x}^* \approx \alpha(\mathbf{x}^{k-1} - \mathbf{x}^*), \quad \alpha < 1 \tag{4}$$

(observe that the empirical observation of (4) led Wainwright [21] to propose an acceleration scheme for linear systems).

Now, if (4) holds, the following approximate equalities will also hold:

$$\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b} \approx \alpha(\mathbf{A}\mathbf{x}^k - \mathbf{b}) \approx \alpha^2(\mathbf{A}\mathbf{x}^{k-1} - \mathbf{b}). \tag{5}$$

But, from (5), we may "deduce" that:

$$1 > \alpha \approx \| \mathbf{A}\mathbf{x}^{k+1} - \mathbf{b} \|_\infty / \| \mathbf{A}\mathbf{x}^k - \mathbf{b} \|_\infty \approx \| \mathbf{A}\mathbf{x}^k - \mathbf{b} \|_\infty / \| \mathbf{A}\mathbf{x}^{k-1} - \mathbf{b} \|_\infty$$

and that the maximum-modulus component of $\mathbf{A}\mathbf{x}^k - \mathbf{b}$ is the same at consecutive iterations.

The observations above justify the tests performed in Step 2 of the algorithm. If these tests are satisfied, we judge that, in fact, the approximate equality (4) holds. Therefore, we perform the classical acceleration Step 3.

Convergence considerations

If the set defined by (2) is nonempty, we know, from [29] that the sequence $(\mathbf{x}^k)$ converges to a point of this set, when $(\mathbf{x}^k)$ is generated by the nonaccelerated version of the method. Algorithm 2.1 may be suitably modified in order that such a result hold also for the accelerated version. One of the possible modifications is the following:

Let p be a positive integer, $0 < \theta < 1$. Set $\bar{\mathbf{F}}_0 = \mathbf{F}_0$. For every k, multiple of p, test the inequality:

$$\| \mathbf{F}_k \| \leqslant \theta \| \bar{\mathbf{F}}_{k-p} \|. \tag{6}$$

If (6) holds, set $\bar{\mathbf{F}}_k = \mathbf{F}_k$, and continue. If (6) does not hold, set $\bar{\mathbf{F}}_k = \mathbf{F}_{k-p}$ and eliminate the acceleration steps for the next p iterations.

It is easy to see that, in this way, we guarantee that $\lim \mathbf{F}_k = 0$. Thus, every accumulation point of $(\mathbf{x}^k)$ is a feasible point of (2). That is, the algorithm will finish with a "Convergence of Type 0" diagnostic.

If the feasible region (2) is empty, the results in [29] guarantee that the nonaccelerated algorithm converges to a point in the prisma, which, or course, does not satisfy $\mathbf{A}\mathbf{x} = \mathbf{b}$. Clearly, with the acceleration procedure and the modification above, the algorithm will keep this property. In fact, after a finite number of iterations the inequality (6) will no longer hold, and, therefore, the algorithm will behave as the nonaccelerated version.

3. NUMERICAL EXPERIMENTS

Consider the following linear programming problem, which arises from a deterministic generation load scheduling application (see [31]).

Maximize

$$z = \sum_{i=n-5}^{n} x_i$$

s.t.

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{l} \leqslant \mathbf{x} \leqslant \mathbf{u},$$

where

$$
\mathbf{A}=\left[\begin{array}{ccccccc}
\mathbf{C} & -\mathbf{I} & & & & & \\
\mathbf{B} & 0 & \mathbf{I} & & & & \\
 & & \mathbf{I} & \mathbf{C} & -\mathbf{I} & & \\
 & -\mathbf{I} & \mathbf{B} & 0 & \mathbf{I} & & \\
\hdashline
 & & & \mathbf{I} & \mathbf{C} & -\mathbf{I} & \\
 & & & -\mathbf{I} & \mathbf{B} & 0 & \mathbf{I}
\end{array}\right],
$$

$$
\mathbf{C}=\begin{bmatrix}
1 & 1 & & & & & & & & & & \\
 & & 1 & 1 & & & & & & & & \\
 & & & & 1 & 1 & & & & & & \\
 & & & & & & 1 & 1 & & & & \\
 & & & & & & & & 1 & 1 & & \\
 & & & & & & & & & & 1 & 1
\end{bmatrix},
$$

$$
\mathbf{B}=\begin{bmatrix}
1 & 1 & & & & & & & & \\
-1 & -1 & 1 & 1 & & & & & & \\
 & -1 & -1 & 1 & & & & & & \\
 & & -1 & -1 & 1 & 1 & -1 & -1 & & \\
 & & & & & & 1 & 1 & -1 & -1 \\
 & & & & & & & & 1 & 1
\end{bmatrix},
$$

$l_1 = (324, 0, 1314.144, 0, 1796.256, 0, 3628.8, 0, 1249.344, 0, 1184.544, 0, 0, 0, 0, 0, 0, 0, 2000, 3350, 7000, 12{,}743, 4400, 900)$,

$u_1 = (2151.36, 10^{30}, 8755.776, 10^{30}, 11{,}982.816, 10^{30}, 22{,}830.336, 10^{30}, 8335.872, 10^{30}, 8125.92, 10^{30}, 10^{30}, 10^{30}, 10^{30}, 10^{30}, 10^{30}, 10^{30}, 5000, 17{,}027, 12{,}500, 21{,}166, 11{,}000, 6150)$,

$\mathbf{l} = (l_1, l_1, \ldots, l_1)$,

$\mathbf{u} = (u_1, u_1, \ldots, u_1)$

$n = 2304.$

$m = 1152.$

We ran this test using the well known simplex-like algorithm MINOS [32, 33] using the VAX-785 of the State University at Campinas. It took 510 CPU seconds to obtain a solution, where $z = 72{,}843$. Only the Phase 1 (Feasibility Phase) of the algorithm used 500 CPU seconds and 1633 iterations.

Then, we considered the following related feasibility problem:

Find $(\mathbf{x}, z)$ such that

$$
\begin{gathered}
\mathbf{A}\mathbf{x} = \mathbf{b}, \quad n = 2304, \quad m = 1152, \\
\sum_{i=n-5}^{n} x_i - z = 0, \\
\mathbf{l} \leqslant \mathbf{x} \leqslant \mathbf{u}, \\
72{,}843 \leqslant z \leqslant 10^{30}.
\end{gathered} \tag{7}
$$

Table 1

t	r = 1	1.15	1.5
−0	3172, 1724	2756, 1503	1850, 1015
0.01	†	77, 59	120, 82
0.1	112, 79	75, 58	273, 167

†$\|\mathbf{F}_k\|_\infty$ reached the value 0.015625 at iteration 131 and couldn't improve it in 500 iterations.

Table 2

t	r = 1	1.15	1.5
−0	131, 86†	123, 81	96, 68
0.01	36, 37	23, 30	33, 35
0.1	28, 32	34, 36	18, 28

†Convergence of type 1, $\|F\| = 0.0206$.

We ran Algorithm 2.1 with $\mathbf{x}^0 = (0, \ldots, 0)$, $p = \infty$, $\epsilon_1 = 10^{-2}$, $\epsilon_2 = 10^{-6}$, and different values for r and t. The results are shown in Table 1. The pair (k, s) means that the method converged using k iterations and s seconds of CPU time.

Replacing the constraint (7) by the less restrictive

$$0 \leqslant z \leqslant 10^{30},$$

we obtain a problem which is closer to the Feasibility Phase of the simplex method when applied to this problem. In this case, we obtain the results which are shown in Table 2.

The last set of experiments with this problem involves an infeasible problem. The constraint (7) was replaced by

$$80{,}000 \leqslant z \leqslant 10^{30} \tag{8}$$

With $r = 1$, the three versions of Algorithm 2.1 converged (in 15, 11 and 8 iterations respectively) to a point where $\|\mathbf{F}_k\|_\infty = 7157$, and the most violated constraint was the last one. The fact that the difference between 80,000 and 7157 is precisely 72,843, the value of the objective function at a solution point of the initial problem, is quite suggestive.

In fact, consider an LP problem of the form:

$$\begin{aligned} &\text{maximize} \quad \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{s.t.} \quad &\mathbf{A}\mathbf{x} = \mathbf{b} \\ &\mathbf{l} \leqslant \mathbf{x} \leqslant \mathbf{u}, \end{aligned} \tag{9}$$

and suppose that the following feasibility problem:

$$\begin{aligned} &\mathbf{A}\mathbf{x} = \mathbf{b}, \\ &\mathbf{c}^{\mathrm{T}}\mathbf{x} = z, \\ &\mathbf{l} \leqslant \mathbf{x} \leqslant \mathbf{u}, \\ &z \geqslant z_0, \end{aligned} \tag{10}$$

is infeasible, and that the SOP method (with $r = 1$) converges to the point $(\mathbf{x}^*, z_0)$, when applied to (8). Moreover, suppose that the only constraint violated by $(\mathbf{x}^*, z_0)$ is precisely $\mathbf{c}^{\mathrm{T}}\mathbf{x} = z$. It follows that $(\mathbf{x}^*, \mathbf{c}^{\mathrm{T}}\mathbf{x}^*)$ is a solution of the problem:

$$\begin{aligned} &\mathbf{A}\mathbf{x} = \mathbf{b}, \\ &\mathbf{c}^{\mathrm{T}}\mathbf{x} = z, \\ &\mathbf{l} \leqslant \mathbf{x} \leqslant \mathbf{u}, \\ &z \geqslant \mathbf{c}^{\mathrm{T}}\mathbf{x}^*. \end{aligned}$$

But $(\mathbf{x}^*, z_0)$ should be the point on the prisma which is closest to the manifold defined by $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{c}^{\mathrm{T}}\mathbf{x} = z$. Therefore, a point $(\bar{\mathbf{x}}, z_0)$ in the prisma such that $\mathbf{c}^{\mathrm{T}}\bar{\mathbf{x}} > \mathbf{c}^{\mathrm{T}}\mathbf{x}^*$ is unlikely to occur.

The observations and experiments led us to the formulation of the following heuristic strategy for solving (9):

Algorithm 3.1

Consider the problem (9). Suppose that it is known that $\mathbf{c}^T\mathbf{x} < c_1$ for all $\mathbf{x}$ belonging to the feasible set. Let η be a (small) positive number. Set $c_2 = c_1$.

Step 1—Apply Algorithm 2.1 to the problem

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$
$$\mathbf{l} \leqslant \mathbf{x} \leqslant \mathbf{u}.$$

If a solution is found, let us call it $\mathbf{x}^1$. Set $c_3 = \mathbf{c}^T\mathbf{x}^1$. Otherwise, stop (problem infeasible).

Step 2—Apply Algorithm 2.1 ($r = 1$) to the problem (10) with $z_0 = c_2$, using $\mathbf{x}^1$ as the initial point. Suppose that the sequence generated by the algorithm converges to $(\mathbf{x}^*, z^*)$. We consider four possibilities:

(i) If $(\mathbf{x}^*, z^*)$ is feasible and $c_2 \geqslant c_1 - \eta$, stop.
(ii) If $(\mathbf{x}^*, z^*)$ is feasible but $c_2 < c_1 - \eta$, set $c_3 \leftarrow c_2$, $c_2 \leftarrow (c_1 + c_2)/2$ and repeat Step 2.
(iii) If $z^* > \mathbf{c}^T\mathbf{x}^*$, set $c_1 \leftarrow c_2$, $c_2 \leftarrow \max\{\mathbf{c}^T\mathbf{x}^*, (c_2 + c_3)/2\}$ and repeat Step 2.
(iv) If $(\mathbf{x}^*, z^*)$ is infeasible, but $\mathbf{c}^T\mathbf{x}^* \approx z^*$, then set $c_1 \leftarrow c_2$, $c_2 \leftarrow (c_2 + c_3)/2$, and repeat Step 2.

With this algorithm, we expect to obtain a point where $\mathbf{c}^T\mathbf{x}$ assumes its maximum possible value up to an accuracy η. We also expect that the possibility (iv) is unlikely to occur. However, we don't have a theoretical justification for this assertion.

We ran Algorithm 3.1 with the following low-dimensional problem [6]:

$$\text{maximize} \quad 3x_1 + x_2$$

s.t.

$$x_1 + 2x_2 \leqslant 3,$$
$$2x_1 + x_2 \leqslant 3,$$
$$2x_1 + 4x_2 \leqslant 6,$$
$$4x_1 + 2x_2 \leqslant 6,$$
$$2x_1 + 2x_2 \leqslant 4.$$

Brocklehurst and Dennis [6] report that Algorithm 350 of ACM [3] the stable LU-type implementation of simplex method of Bartels and Golub, "iterated indefinitely" when applied to this problem. We used Algorithm 3.1 beginning with $c_1 = 100$. The progress of the algorithm is shown in Table 3.

4. CONCLUSIONS

In this paper we introduced a new acceleration procedure for the successive orthogonal projection (SOP) method applied to linear feasibility problems. The method behaved quite well in some large scale real life problems. We outlined an algorithm, based in the iterative use of the

Table 3

Step	Iterations	Type of convergence	c_1	c_2
1	1	0	—	−1000
2	7	1	100	100
3	10	1	100	28.88414
4	8	1	28.88414	11.00476
5	7	1	11.00476	6.509675
6	6	1	6.509675	5.379556
7	5	1	5.379556	5.0954326
8	5	1	5.0954346	5.024004
9	4	1	5.024004	5.00603755
10	4	1	5.00603755	5.001522379
11	1	0	5.001522379	5.000383692

previous one, for handling general large scale LP problems. Some questions concerning this algorithm, as well as its possible improvements, are to be answered in future research.

Acknowledgements—The author acknowledges stimulating discussions with Marcia Ap. Gomes Ruggiero, Ana Friedlander and Antonio Carlos Moretti. We also acknowledge the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) and to the CNPQ for financial support given to this work.

REFERENCES

1. P. Gill, W. Murray and M. H. Wright, *Practical Optimization.* Academic Press, New York (1981).
2. D. G. Luenberger, *Linear and Nonlinear Programming.* Addison-Wesley, Reading, MA (1984).
3. R. H. Bartels and G. H. Golub, The simplex method of linear programming using LU decomposition. *Commun. ACM* **12,** 266–268 (1969).
4. G. B. Dantzig, *Linear Programming and Extensions.* Princeton University Press, Princeton, NJ (1963).
5. L. M. Bregman, The relaxation method of finding a common point of convex sets and its application to the solution of problems in convex programming. *U.S.S.R. comp. Math. math. Phys.* **7,** 200–217 (1967).
6. E. R. Brocklehurst and K. Dennis, A comparison of six algorithms for dense linear programs. NPL Report NAC 51 (1974).
7. Y. Censor, Finite series expansion reconstruction methods. *Proc. IEEE* **71,** 409–419 (1983).
8. Y. Censor and G. T. Herman, On some optimization techniques in image reconstruction from projections. Medical Image Processing Group Technical Report No. MIPG107, Department of Radiology, University of Pennsylvania (1986).
9. A. R. De Pierro and A. N. Iusem, A simultaneous projection method for linear inequalities. *Linear Algebra Applic.* **64,** 243–253 (1985).
10. R. Gordon, R. Bender and T. Herman, Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *J. theor. Biol.* **29,** 471–481 (1970).
11. G. T. Herman, *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography.* Academic Press, New York (1980).
12. S. Kaczmarz, Angenaerte Auflösung von Systemen linearer Gleichungen. *Bull. Acad. Polon. Sci. Lett.* **A35,** 355–357 (1937).
13. J. E. Spingarn, A primal–dual projections method for solving systems of linear inequalities. *Linear Algebra Applic.* **65,** 45–62 (1985).
14. Y. Censor, Row-action methods for huge and sparse systems and their applications. *SIAM Rev.* **23,** 444–466 (1981).
15. U. Garcia-Palomares, Q-relaxation method for solving a system of linear inequalities. TM No. 19, Mathematics and Computed Science Division, Argonne National Laboratory, IL (1983).
16. J. M. Martinez, The projection method for solving nonlinear systems of equations under the most violated constraint condition control. *Comput. Math. Applic.* **11,** 987–993 (1985).
17. J. M. Martínez, Solution of nonlinear systems of equations by an optimal projection method. *Computing* **37,** 59–70 (1986).
18. J. M. Martínez, Solving systems of nonlinear equations by means of an accelerated successive orthonormal projections method. *J. Comput. appl. Math.* **16,** 169–179 (1986).
19. J. M. Martínez and R. J. B. Sampaio, Parallel and sequential Kaczmarz methods for solving underdetermined nonlinear equations. *J. Comput. appl. Math.* **15,** 169–179 (1986).
20. K. H. Meyn, Solution of underdetermined nonlinear equations by stationary iteration methods. *Numer. Math.* **42,** 161–172 (1983).
21. R. L. Wainwright, Four-dimensional x-projection method (with acceleration techniques) for solving systems of linear equations. *Comput. Math. Applic.* **8,** 329–337 (1982).
22. A. C. Moretti, Aplicação do método de projeções ortogonais sucessivas a um problema de controle. Tese de Mestrado, IMECC–UNICAMP (1985).
23. T. F. Coleman, *Large Sparse Numerical Optimization. Lecture Notes in Computer Science,* Vol. 165, Springer, Berlin (1984).
24. A. George and M. T. Heath, Solution of sparse linear least squares using Givens rotations. *Linear Algebra Applic.* **34,** 69–83 (1980).
25. A. George, M. T. Heath and E. Ng, A comparison of some methods for solving sparse least squares problems. *SIAM Jl Sci. Statist. Comput.* **4,** 177–187 (1983).
26. M. T. Heath, Numerical methods for large sparse linear least squares problem. *SIAM Jl Sci. Statist. Comput.* **5,** 497–513 (1984).
27. N. Karmarkar, A new polynomial-time algorithm for linear programming. *Combinatorica* **4,** 373–395 (1984).
28. I. J. Lustig, A practical approach to Karmarkar's algorithm. TR SOL 85-5, Department of Operations Research, Stanford University, CA (1985).
29. L. G. Gubin, B. T. Polyak and E. V. Raik, The method of projections for finding the common point of convex sets. *U.S.S.R. comp. Math. math. Phys.* **7,** 1–24 (1967).
30. N. Gastinel, *Linear Numerical Analysis.* Hermann, Paris (1970).
31. A. Friedlander, Otimization com estrutura escada nas restrições. Tese de Doutorado, FEE UNICAMP (1986).
32. B. A. Murtagh and M. A. Saunders, Large scale linearly constrained optimization. *Math. Prog.* **14,** 41–22 (1978).
33. B. A. Murtagh and M. A. Saunders, MINOS: a large-scale nonlinear programming system (for problems with linear constraints). TR SOL 77-9, Department of Operations Research, Stanford University, CA (1977).